INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-26 06:11 UTC

Indirect Prompt Injection Attacks Targeting LLM-Powered AI Agents Observed in the Wild

SECURITY ANALYSIS | HIGH | CVSS 8.1

SCC Item ID	SCC-STY-2026-0086
Type	Security Analysis
Severity	HIGH
CVSS Base Score	8.1
Affected Products	LLM-powered AI agents and AI-driven applications (vendor-agnostic; any system processing untrusted external content via an LLM agent)
Published	2026-04-24
Discovery Source	Gemini

Executive Summary

Researchers have documented real-world exploitation of indirect prompt injection (IPI), an attack class that embeds adversarial instructions inside content that AI agents retrieve and execute as trusted commands, without any action required from the end user. Confirmed objectives include financial fraud, credential theft, and unauthorized data exfiltration, meaning organizations that have deployed AI agents to automate workflows have an attack surface they likely have not yet inventoried or modeled. This is not a theoretical concern: OWASP has designated prompt injection the top risk for LLM applications, and the absence of any standardized input sanitization layer between external content and agent instruction processing means the exposure is structural, not incidental.

Technical Analysis

Indirect prompt injection occupies a distinct position in the AI attack taxonomy. Unlike direct prompt injection, where an attacker interacts with an LLM directly, IPI requires no privileged access to the target system. The attacker plants adversarial instructions in content the agent will retrieve autonomously, a web page the agent browses, a document it summarizes, an email it processes, or an API response it parses. The agent, treating retrieved content as part of its operating context, executes the embedded instructions as if they were legitimate task directives.

Researchers have now moved this attack from proof-of-concept into confirmed real-world exploitation. The attack surface is defined by agent capability, not by vendor: any agentic system with access to external content sources is exposed. That includes agents integrated into enterprise productivity suites, customer service

automation, security tooling, and software development assistants.

Concealment techniques observed in the wild include white-on-white text, CSS-hidden elements, and zero-width Unicode characters, methods specifically designed to ensure the injected payload is invisible to human reviewers while remaining fully parseable by the LLM. This asymmetry is operationally significant: a security analyst reviewing flagged content may see nothing, while the agent has already acted on hidden instructions.

MITRE ATT&CK provides useful framing. T1059 (Command and Scripting Interpreter) maps to the core execution mechanism: the agent functions as an interpreter, and the adversary-controlled instructions are the script. T1566 (Phishing) maps to the delivery vector, since malicious content reaches the agent through channels analogous to socially engineered lures. T1027 (Obfuscated Files or Information) maps to the concealment layer. T1530 (Data from Cloud Storage) is relevant where agents have access to cloud-resident data that becomes an exfiltration target once the agent is compromised.

The underlying weaknesses are CWE-20 (Improper Input Validation) and CWE-116 (Improper Encoding or Escaping of Output). No standardized sanitization layer exists between external content ingestion and LLM instruction processing, the agent architecture itself creates the vulnerability. OWASP LLM01 designates prompt injection, of which indirect prompt injection is a critical variant, as the top risk category for LLM applications. No CVE has been assigned because IPI is a vulnerability class, not a discrete flaw in a versioned product, which also means no patch is forthcoming from any single vendor.

Attribution to specific threat actors remains low confidence. No named group has been publicly confirmed in connection with observed IPI campaigns. The barrier to entry, however, is low: the attack requires no zero-day, no malware, and no network intrusion, only the ability to place content where an agent will retrieve it.

Action Checklist

1. Step 1: Assess exposure, inventory every AI agent or LLM-powered workflow in your environment that retrieves or processes external content (web pages, documents, emails, API responses); treat each as a potential IPI attack surface regardless of vendor
2. Step 2: Review controls, evaluate whether output validation exists between agent-retrieved content and instruction execution; assess whether agents operate with least-privilege permissions, particularly for actions involving credential access, data access, financial systems, or external communications
3. Step 3: Update threat model, add indirect prompt injection as an explicit threat scenario in your AI risk register; map it to T1059, T1566, T1027, and T1530 in your ATT&CK-aligned detection framework and document which agent deployments have no current detection coverage
4. Step 4: Communicate findings, brief engineering and product leadership on which AI agent deployments are exposed; frame the risk in terms of agent-accessible data and actions, not abstract AI safety concepts, and establish ownership for remediation per deployment
5. Step 5: Monitor developments, track OWASP LLM Top 10 updates, NIST AI Risk Management Framework (NIST AI RMF 1.0, particularly the GOVERN and MAP functions), and vendor-specific mitigations from your AI platform providers; no industry-wide technical standard for IPI prevention currently exists, so organizational controls and architecture decisions carry the full defensive burden

IR / Forensic Enrichment

Triage Priority	URGENT

Escalation Criteria	Escalate immediately to CISO and legal counsel if agent execution logs show evidence of IPI-triggered actions — specifically: unauthorized external data transmissions from agent processes, credential access events during agent execution windows not initiated by a human user, or financial API calls (payment processors, banking integrations) correlated with agent processing of external content — as these constitute confirmed IPI exploitation and may trigger breach notification obligations under GDPR, CCPA, or PCI-DSS depending on data types involved.
Recovery Notes	After containing an IPI incident, revoke and rotate all API keys and OAuth tokens held by the compromised agent before redeployment, and audit the full execution log for the incident window to enumerate every tool invocation the agent performed under adversarial instruction — do not assume only the final observed action was taken. Redeploy agents only after implementing architectural controls: human-in-the-loop confirmation for high-risk tool categories (send, write, delete, external transmission), explicit untrusted-data framing in system prompts, and output filtering for instruction-pattern detection using a secondary LLM call or rule-based classifier. Monitor redeployed agents with elevated logging verbosity for a minimum of 30 days post-recovery, specifically watching for recurrence of anomalous tool invocation sequences that correlate with external content retrieval events.
Forensic Artifacts	LLM API request/response logs (stored at the application layer or via a proxy such as LiteLLM or a custom logging middleware): examine the 'messages' array in API payloads for assistant turns that contain imperative instructions sourced from retrieved external content rather than from the system prompt — this is the primary forensic record of IPI payload delivery and agent instruction override | Agent tool call invocation logs: in LangChain these appear as 'Action' and 'Action Input' entries in agent executor output; in OpenAI Assistants API they appear as 'tool_calls' objects in run step details — cross-reference tool invocations against the external content retrieval event that immediately preceded them to reconstruct the IPI kill chain | Cloud IAM and SaaS audit logs correlated with agent execution timestamps: AWS CloudTrail S3 GetObject/PutObject events, Microsoft Graph audit logs for mail send or file access events, Google Workspace Admin audit logs for Drive file access or Gmail send events — look for actions performed under the agent's service account identity during processing windows that correlate with external content retrieval, as these represent the downstream impact of successful IPI execution | Retrieved content cache or fetch logs: HTTP access logs from the agent's web retrieval tool, email body content processed by mail-reading integrations, or document content passed to the LLM — examine for obfuscation techniques specific to IPI: Unicode directional override characters (U+202E), zero-width joiners (U+200D), HTML elements with display:none or visibility:hidden styling, or white-text instructions embedded in documents that are invisible to human reviewers but present in the text the LLM processes | Network flow logs for the agent execution environment: capture outbound connections from the agent process or container during the incident window — IPI attacks targeting data exfiltration will produce anomalous outbound HTTP POST or SMTP connections to attacker-controlled infrastructure; compare destination IPs and domains against threat intelligence feeds and against the agent's known-good external communication whitelist

Per-Action IR Details

Step 1: Assess exposure — inventory every AI agent or LLM-powered workflow in your environment that retrieves or processes external content (web pages, documents, emails, API responses); treat each as a potential IPI attack surface regardless of vendor

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Establishing IR capability and asset inventory as foundational readiness for detecting and responding to novel attack classes against AI agent infrastructure

Controls: NIST IR-4 (Incident Handling) — requires an incident handling capability that includes preparation; AI agent deployments processing external content must be scoped into the IR plan, NIST RA-3 (Risk Assessment) — assess risk associated with each LLM agent's external content retrieval surface, including tool-call permissions and data access scope, NIST SA-9 (External System Services) — AI platform APIs (OpenAI, Anthropic, Azure OpenAI, Google Vertex) constitute external services; their agent execution environments must be inventoried and risk-assessed, CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory) — extend asset inventory to include LLM agent deployments, their tool integrations, external data sources they query, and the identity context under which they execute, CIS 2.1 (Establish and Maintain a Software Inventory) — catalog all AI agent frameworks in use (LangChain, AutoGen, CrewAI, custom orchestration layers) and their versions, as IPI exploitability varies by how each framework handles retrieved content

Compensating: Run a two-person manual discovery sprint: (1) Query your CI/CD pipeline configs and environment variable stores for API keys referencing LLM providers (grep -r 'OPENAI_API_KEY\|ANTHROPIC_API_KEY\|AZURE_OPENAI' /etc /opt /srv 2>/dev/null). (2) Search deployed container images and serverless function configs for agent framework imports (grep -r 'langchain\|autogen\|crewai\|openai.agents' across your IaC and app repos). (3) Interview engineering leads using a structured questionnaire: does this service fetch external URLs, parse uploaded documents, read emails, or call external APIs and pass results to an LLM? Document each finding in a shared spreadsheet with columns: agent name, LLM provider, external content sources, tool permissions, data access scope.

Evidence: Before scoping, preserve a point-in-time snapshot of: (1) All outbound API call logs to LLM provider endpoints (api.openai.com, api.anthropic.com, *.openai.azure.com) from your network proxy or firewall — retain at minimum 90 days. (2) Application logs showing tool invocations or function-call payloads from agent orchestration layers — these will contain the raw external content the agent retrieved and processed, which is the forensic ground truth for whether IPI payloads transited your environment. (3) Cloud IAM audit logs (AWS CloudTrail, Azure Activity Log, GCP Audit Log) showing what permissions the agent service accounts hold — this establishes blast radius before you begin containment decisions.

Step 2: Review controls — evaluate whether output validation exists between agent-retrieved content and instruction execution; assess whether agents operate with least-privilege permissions, particularly for actions involving credential access, data access, financial systems, or external communications

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Evaluating the adequacy of preventive controls as a precondition for effective incident response; identifying control gaps before an IPI-triggered execution event occurs

Controls: NIST SI-10 (Information Input Validation) — LLM agent pipelines must validate that retrieved external content (web pages, documents, email bodies) is treated as untrusted data, not as executable instruction; absence of this control is the direct enabler of IPI, NIST AC-6 (Least Privilege) — agent service accounts and tool integrations must be scoped to minimum necessary permissions; an agent capable of reading email AND sending email AND accessing credential stores simultaneously represents a critical over-permissioned attack surface, NIST AC-3 (Access Enforcement) — enforce access controls on what actions agents can invoke autonomously versus what requires human confirmation, particularly for financial transactions, credential retrieval, and external data transmission, NIST SI-7 (Software, Firmware, and Information Integrity) — implement prompt integrity controls where feasible; verify that system prompts governing agent behavior have not been overridden or appended by retrieved content, CIS 3.3 (Configure Data Access Control Lists) — apply need-to-know data access to agent identities; an agent automating calendar scheduling has no legitimate need for access to credential vaults or payment APIs, CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts) — agent service accounts must not hold administrative privileges; map each agent's OAuth scopes, API keys, and IAM roles against the principle of least function

Compensating: Without a dedicated AI security tool, perform manual control gap analysis: (1) For each agent identified in Step 1, extract its system prompt and tool definitions — in LangChain this is in the agent executor config; in AutoGen it is in the AssistantAgent system_message parameter. Inspect whether the system prompt explicitly instructs the agent to treat retrieved content as data, not commands. (2) Enumerate all tool integrations and their permission scopes using CLI: for AWS, run 'aws iam simulate-principal-policy' against the agent's role; for Azure, use 'az role assignment list --assignee '. (3) For agents using OAuth tokens (e.g., Gmail, Slack, Microsoft Graph), audit granted scopes at the OAuth app registration level — document any scope that includes send, write, delete, or admin.

Evidence: Capture before control review: (1) Agent system prompt text as deployed in production — if prompts are stored in environment variables or a prompt management service, export and hash them (sha256sum) to establish a baseline for detecting future IPI-driven prompt manipulation. (2) Tool schema definitions (function-call JSON schemas in OpenAI format, or tool descriptions in other frameworks) — these define what actions the agent can take and are the forensic basis for assessing what an IPI attacker could have commanded. (3) OAuth token grant records and API key issuance logs showing when agent credentials were created, what scopes were granted, and whether any scope elevation has occurred since initial deployment.

Step 3: Update threat model — add indirect prompt injection as an explicit threat scenario in your AI risk register; map it to T1059, T1566, T1027, and T1530 in your ATT&CK-aligned detection framework and document which agent deployments have no current detection coverage

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Incorporating newly documented threat scenarios into IR planning artifacts; updating detection frameworks to reflect confirmed real-world exploitation of IPI by threat actors pursuing financial fraud and credential theft

Controls: NIST IR-8 (Incident Response Plan) — update the IR plan to include IPI as a named threat scenario with specific indicators, affected agent deployments, and designated response owners; generic AI incident handling is insufficient given IPI's confirmed exploitation for financial fraud and credential theft, NIST RA-3 (Risk Assessment) — formally document IPI in the risk register with likelihood informed by Google and Forcepoint research confirming in-the-wild exploitation; assess impact based on each agent's accessible data and action scope, NIST SI-4 (System Monitoring) — establish monitoring requirements specific to IPI: anomalous tool invocations, unexpected external data transmissions initiated by agent processes, and credential access events correlating with agent execution windows, CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — extend the vulnerability management process to cover AI-specific weaknesses including OWASP LLM Top 10 LLM01 (Prompt Injection); IPI has no CVE but carries confirmed CVSS 8.1 severity and active exploitation, CIS 8.2 (Collect Audit Logs) — ensure audit logging is enabled for all agent tool invocations, LLM API calls, and downstream actions triggered by agent execution; without this, IPI-triggered actions are forensically invisible

Compensating: Map IPI to ATT&CK manually with two analysts: T1059 (Command and Scripting Interpreter) — the injected prompt functions as a command issued to the agent's execution environment; T1566 (Phishing) — the adversarial instruction is embedded in content the agent retrieves (a web page, a document, an email), analogous to a phishing payload delivered through a trusted channel; T1027 (Obfuscated Files or Information) — IPI payloads are frequently obfuscated using invisible Unicode characters, HTML comment injection, or white-text-on-white-background techniques to hide instructions from human reviewers while remaining visible to the LLM; T1530 (Data from Cloud Storage Object) — agents with access to cloud storage or SaaS data repositories can be directed by IPI to exfiltrate specific files. Write Sigma rules for each: a baseline Sigma rule for T1530 via IPI should alert on agent process identities initiating bulk object GET requests from S3, SharePoint, or Google Drive outside normal operational hours.

Evidence: Before updating the threat model, collect: (1) Historical LLM API request/response logs — specifically look for responses containing imperative natural-language instructions directed at the agent (phrases like 'ignore previous instructions', 'forward this to', 'retrieve and send') embedded within otherwise normal retrieved content; these are forensic indicators of IPI attempts whether or not they succeeded. (2) Agent tool call logs showing unexpected invocations — e.g., a document-summarization agent that suddenly invokes a send_email or http_post tool during processing of an external document is a high-fidelity IPI execution indicator. (3) Obfuscation artifact samples: search retrieved document caches or web fetch logs for Unicode directional override characters (U+202E, U+200B through U+200F), HTML tags containing display:none or color:white styling, or zero-width space characters (U+FEFF) that could carry hidden instructions.

Step 4: Communicate findings — brief engineering and product leadership on which AI agent deployments are exposed; frame the risk in terms of agent-accessible data and actions, not abstract AI safety concepts, and establish ownership for remediation per deployment

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Establishing organizational roles, communication structures, and ownership accountabilities as prerequisites for coordinated response to IPI incidents affecting AI agent deployments

Controls: NIST IR-6 (Incident Reporting) — establish internal reporting channels and escalation paths specific to IPI incidents; because IPI may trigger financial transactions or credential exfiltration autonomously, the reporting timeline must account for agent execution speed — human review may occur after harm is complete, NIST IR-7 (Incident Response Assistance) — identify which engineering teams own each agent deployment and can perform emergency containment (disabling tool integrations, revoking API keys, toggling agent off) within a defined SLA, NIST IR-4 (Incident Handling) — assign explicit IR roles for AI agent incidents: who declares an IPI incident, who has authority to disable a production agent, who communicates with affected business units when an agent has been manipulated into performing unauthorized financial or data actions, NIST IR-2 (Incident Response Training) — brief engineering and product owners using concrete IPI scenarios drawn from the Google and Forcepoint research: an agent reading a malicious web page and autonomously forwarding sensitive emails, or an agent processing an adversarial document and initiating unauthorized API calls to payment systems, CIS 7.2 (Establish and Maintain a Remediation Process) — assign remediation ownership per agent deployment with documented timelines; agents with access to financial systems or credential stores require immediate remediation owners with authority to act

Compensating: Produce a one-page risk brief for each high-risk agent deployment using this structure: (1) Agent name and function. (2) External content sources it retrieves (specific URLs, document types, email accounts, API endpoints). (3) Actions it can take autonomously (list each tool integration and what it does in plain language). (4) Worst-case IPI scenario: if an adversary embeds instructions in a [web page / document / email] this agent processes, they could cause the agent to [specific action — e.g., forward all emails in the connected mailbox to an external address, initiate a payment via the connected Stripe API, retrieve and exfiltrate files from the connected SharePoint site]. (5) Named remediation owner and their contact. Deliver briefings synchronously — do not rely on async documentation for findings at this severity level.

Evidence: Before briefing leadership, preserve: (1) A current export of each agent's active tool integrations and their permission scopes — this is the evidence base for the 'what can this agent do' section of the risk brief and must reflect production state at briefing time, not documentation that may be stale. (2) Any existing agent execution logs showing the volume and diversity of external content the agent has processed — this quantifies historical exposure and informs whether a retroactive compromise review is warranted. (3) Incident ticket or risk register entry documenting the date IPI was identified as an active threat in your environment — establishes the organizational awareness timeline, which is relevant if an IPI-triggered breach subsequently requires regulatory notification.

Step 5: Monitor developments — track OWASP LLM Top 10 updates, NIST AI RMF guidance, and vendor-specific mitigations from your AI platform providers; no industry-wide technical standard for IPI prevention currently exists, so organizational controls and architecture decisions carry the full defensive burden

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Incorporating threat intelligence on IPI into organizational learning, updating detection capabilities, and improving preventive controls as the IPI threat landscape and available mitigations evolve

Controls: NIST SI-5 (Security Alerts, Advisories, and Directives) — establish a formal process for receiving and acting on IPI-relevant advisories from OWASP (LLM Top 10), NIST AI RMF working groups, and AI platform vendors (OpenAI safety bulletins, Anthropic model card updates, Google DeepMind security advisories), NIST IR-4 (Incident Handling) — update incident handling procedures as IPI detection and prevention techniques mature; current absence of an industry technical standard means organizational procedures must be revisited quarterly as OWASP LLM Top 10 and NIST AI RMF guidance is revised, NIST AU-6 (Audit Record Review, Analysis, and Reporting) — establish a recurring review cadence for agent execution logs specifically looking for IPI indicators: anomalous tool invocation sequences, unexpected external transmission events, and retrieved content containing imperative instruction patterns, NIST RA-3 (Risk Assessment) — reassess AI agent risk posture when AI platform vendors release architectural changes affecting how retrieved content is processed (e.g., changes to function-calling trust boundaries, context window handling, or tool-use confirmation requirements), CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — include OWASP LLM Top 10 LLM01 tracking in the vulnerability management process; assign a named owner to monitor vendor-specific IPI mitigations (e.g., OpenAI's upcoming prompt shielding features, Microsoft's Prompt Shields in Azure AI Content Safety, Google's equivalent controls in Vertex AI Agent Builder), CIS 8.2 (Collect Audit Logs) — expand audit log collection to cover agent-specific telemetry as vendors release it; monitor vendor roadmaps for native IPI

detection capabilities and integrate into SIEM or log aggregation when available

Compensating: For a two-person team with no enterprise tooling: (1) Subscribe to OWASP LLM Top 10 GitHub releases (github.com/OWASP/www-project-top-10-for-large-language-model-applications) and configure a GitHub notification or RSS feed for new releases. (2) Subscribe to your AI platform provider's security bulletins via email or RSS — OpenAI publishes security updates at openai.com/security; Microsoft Azure AI publishes via the Azure Updates feed filtered on 'AI + Machine Learning'. (3) Set a quarterly calendar reminder to review NIST AI RMF documentation at airc.nist.gov and check for new AI RMF profiles or playbooks relevant to agentic systems. (4) Write a weekly cron job or scheduled task that queries your agent execution logs for IPI indicator patterns using grep or jq: search for tool invocation events where the triggering content source is external (not internal system prompt) and the invoked tool involves data transmission or credential access — pipe results to a daily digest email.

Evidence: Preserve on an ongoing basis for IPI threat monitoring: (1) A versioned archive of each agent's system prompt as deployed — store with git or equivalent so prompt drift (whether from IPI manipulation or engineering changes) is detectable by diff. (2) Vendor changelog records for your AI platform's model versions — model updates can change how injected instructions are processed, expanding or contracting IPI attack surface; correlate model version changes with any anomalies in agent behavior logs. (3) A running log of external threat intelligence on IPI techniques from Google Project Zero, Forcepoint research publications, and OWASP LLM working group outputs — this intelligence log is the evidence base for future risk register updates and IR plan revisions.

Detection Guidance

Detection for IPI is non-trivial because the attack exploits the agent's normal retrieval workflow, there is no malware to signature and no network anomaly to baseline. Hunting and monitoring should focus on behavioral anomalies in agent output and action logs rather than content inspection alone.

Log targets: Agent action logs (tool calls, API invocations, file access, outbound communications initiated by the agent), LLM input/output logging where available, data access logs for systems the agent can reach (cloud storage, email, internal APIs).

Behavioral patterns to hunt for: Agent actions that were not initiated by a human user request and cannot be traced to an explicit task directive; outbound data transfers or API calls immediately following external content retrieval; agent requests to access credentials, authentication tokens, or privileged resources outside the defined task scope; sequences where content retrieval precedes an action the agent has not performed in baseline operation.

Content-layer indicators: Presence of white-on-white text, CSS visibility manipulation, or zero-width Unicode characters (U+200B, U+FEFF, and related code points) in content retrieved by agents; this requires logging raw retrieved content before rendering, which most deployments do not do by default.

Policy gaps to audit: Whether agents operate with permissions scoped to minimum required actions; whether human-in-the-loop checkpoints exist before agent-initiated actions affecting financial systems, credential stores, or external data transmission; whether any output validation layer exists to assess agent-generated actions before execution.

Note: No specific IOC hashes, domains, or IPs associated with confirmed IPI campaigns have been published in the source material for this story. Detection is behavioral and architectural, not indicator-based.

Indicators of Compromise

Type	Value	Context	Confidence
TOOL	`Pending — refer to Google Security and Forcepoint research publications for published indicators`	No specific IOC values (hashes, domains, IPs) were published in the source material provided. Google and Forcepoint researchers documented behavioral patterns and concealment techniques; any campaign-specific indicators would be available in their full research disclosures.	LOW

Framework Mappings

MITRE-ATTACK

- **T1059** — Command and Scripting Interpreter
- **T1530** — Data from Cloud Storage
- **T1027** — Obfuscated Files or Information
- **T1566** — Phishing

NIST-800-53R5

- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **AT-2** — Literacy Training and Awareness
- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **SI-8** — Spam Protection
- **SI-10** — Information Input Validation

OWASP-TOP10-2021

- **A03:2021** — Injection

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **6.3** — Require MFA for Externally-Exposed Applications
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks
- **8.2** — Collect Audit Logs

ISO-27001-2022

- **A.8.26** — Application security requirements
- **A.8.8** — Management of technical vulnerabilities
- **A.5.34** — Privacy and protection of personal information

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication
- **164.308(a)(5)(i)** — Security Awareness and Training
- **164.308(a)(6)(ii)** — Response and Reporting

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures
- **CC7.4** — Responds to identified security incidents

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored
- **DE.AE-08** — Incidents are declared when adverse events meet the defined incident criteria

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1059	Command and Scripting Interpreter	Execution
T1530	Data from Cloud Storage	Collection
T1027	Obfuscated Files or Information	Defense-Evasion
T1566	Phishing	Initial-Access

Sources

Source	URL	Tier
Unveiling AI Agent Vulnerabilities Part I: Introduction to AI ...	https://www.trendmicro.com/vinfo/us/security/news/threat-landscape/...	T3
Threats in LLM-Powered AI Agents Workflows	https://arxiv.org/html/2506.23260v2	T2
OWASP Top 10 for Large Language Model Applications	https://owasp.org/www-project-top-10-for-large-language-model-appli...	T3
The Hidden Attack Surface of LLM-Powered Applications	https://brightsec.com/blog/the-hidden-attack-surface-of-llm-powered...	T3
The Dark Side of LLMs: Agent-based Attacks for Complete ...	https://arxiv.org/html/2507.06850v3	T2

DISCLAIMER

This intelligence report is produced by Tech Jacks Solutions Security Command Center (SCC) for informational purposes only. It does not constitute professional security advice, legal counsel, or an incident response engagement. The information herein is derived from publicly available sources and AI-assisted analysis; while every effort is made to ensure accuracy, Tech Jacks Solutions makes no warranties regarding completeness or timeliness. Organizations should conduct their own validation and consult qualified security professionals before taking action based on this report. Tech Jacks Solutions is not liable for any damages resulting from the use of this information.

Generated 2026-04-26 06:11 UTC by TJS Security Command Center